


**System of Systems Architecture
Evaluation with Concurrent
Development**

SATURN 2007
Mike Gagliardi, Bill Wood

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

 **Software Engineering Institute** | CarnegieMellon © 2007 Carnegie Mellon University

Problem

Severe integration and operational problems can arise due to inconsistencies in addressing system quality attributes between system and software architecture.

This is further exacerbated in a System of Systems (SoS) context where major system and software elements are developed concurrently and independently by a variety of contractors.

Need a methodology to identify these inconsistencies within a system of systems context early in the life cycle.

 **Software Engineering Institute** | CarnegieMellon Gagliardi, Wood
© 2007 Carnegie Mellon University 2

Our Intentions

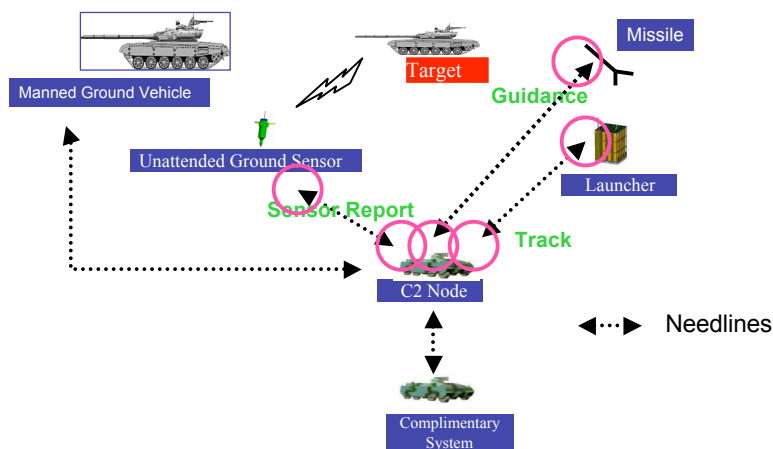
Develop a method based on but not restricted by ATAM and QAW

- Quality attributes, augmented mission threads, and scenarios
- Multi-layered evaluation
 - first pass looking for inconsistencies, also identifying potential systems for further, detailed evaluation (using S-ATAM or ATAM) techniques.
 - Subsequent evaluations of individual systems use mission thread fragments, scenarios from first pass evaluation. May need further decomposition of mission threads, scenarios, etc for “SoS to System” transition.

Work with large-scale programs and contractors to develop techniques and participate in SoS evaluations.



SoS Architecture View (OV-2)



Specific Mission Thread

An enemy tank platoon is threatening a lightly protected company, and comes into the field of view of a UGS, which connects to and informs a MGVC2. An MGVC2 identifies the enemy tanks. The MGVC2 assigns an unmanned missile launcher (UML) to engage the tank platoon. The UML engages and destroys the enemy. The MGVC2 determines that the threat has been eliminated from subsequent UGS signals.



Quality Attribute Augmentations

The engagement will be started within 10 seconds of sensing the enemy tank platoon

Any one failure does not impact any timelines

- Hardware, software, user, vehicle

This happens in an environment where there are 2000 BSEs being tracked

The security level between the MGVC2 and the UML is top secret

Include some measure of (importance, difficulty)



Evaluation

Evaluate high level SoS representations and guidelines

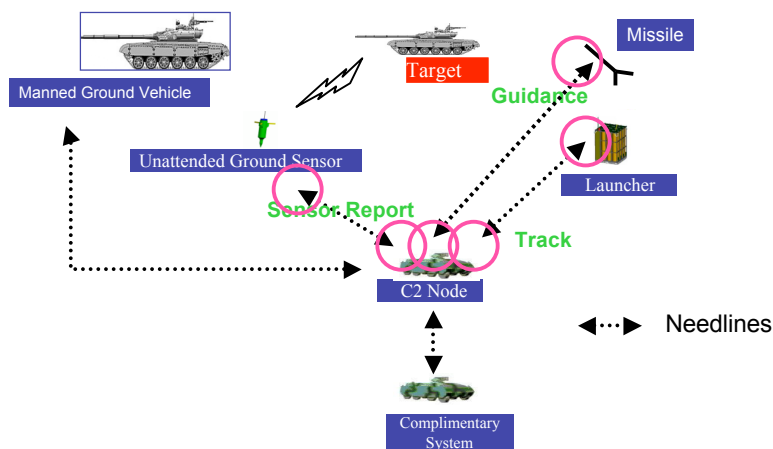
- Cross system mission threads augmentation with quality attributes and high level architecture from different systems
- Only evaluate the areas where the systems interact
- Follow general ATAM steps with key stakeholders; architect walking through architecture using augmented mission threads; evaluation team probing architecture.
- Define gaps and risks and use them to target “focused evaluations” or ATAMs using the specific augmented mission threads for further evaluation
- *Assumes the documentation of SoS architecture guidelines and representations*
- *Use “Mission Thread Augmentation Workshop” early in life cycle*

Evaluate individual systems (or combinations)

- Derive scenarios from the augmented mission threads
- Use “focused evaluations” or ATAMs



SoS Architecture View (OV-2)



Concurrent Development

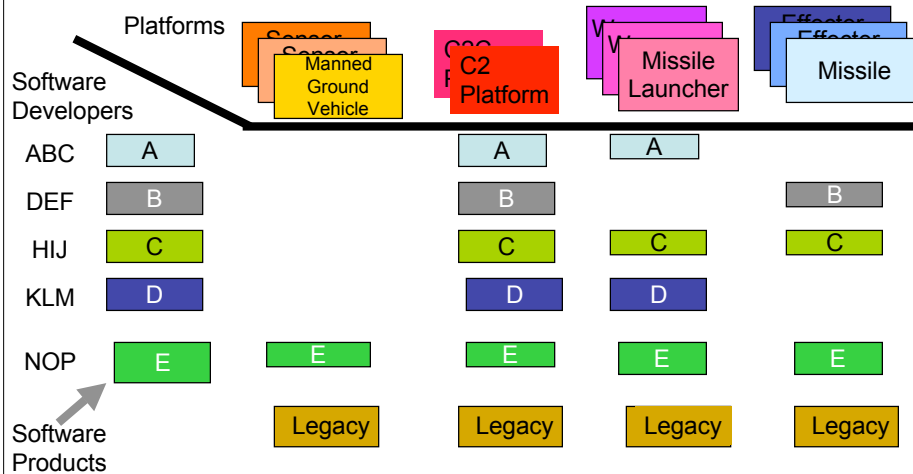
SoS Architecture Guidelines are concurrently developed with some architecture components (views)

- Some folks “get ahead of the guidelines”
- Some legacy components may not be re-engineered to guidelines

Systems are architected and designed concurrently and independently, often with different views and tools. Must reconcile different architecture products as they become available.



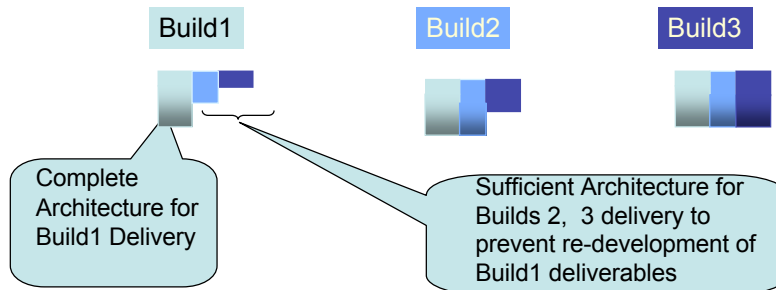
Example: Multiple Contractors



Life-Cycle – Multiple Builds

Evaluation should focus on “Next Build”

- Including enough architecture of future build to minimize extensive re-work between builds (especially safety, security, fault tolerance)



Quality Attribute Augmentations (with builds)

The engagement will be started within 10 seconds of sensing the enemy tank platoon **Phase 1**

Any one failure does not impact any timelines

- Hardware, software, user, vehicle – **Phase 2**

This happens in an environment where there are 2000 BSEs being tracked (1000 in **phase 1**)

The security level between the MGVC2 and the UML is top secret - **phase 3**

Include some measure of (importance, difficulty)



Evaluation (with builds)

Evaluate high level SoS representations and guidelines

- Cross system mission threads augmentation with quality attributes and high level architecture from different systems
- Only evaluate the areas where the systems interact
- Follow general ATAM steps with key stakeholders; architect walking through architecture using augmented mission threads; evaluation team probing architecture.
- Define gaps and risks and use them to target "focused evaluations" or ATAMs using the specific augmented mission threads for further evaluation
- Evaluate each build
- Evaluation focuses on current and next build
 - Including enough architecture of future build to minimize extensive re-work between builds (especially safety, security, fault tolerance)

Evaluate individual systems (or combinations)

- Derive scenarios from the augmented mission threads
- Use "focused evaluations" or ATAMs



Challenges

Technical

- SOA, Enterprise Architecture, legacy / COTS / GOTS components, Massive Middleware products

Evaluation

- Each contractor may use diverse DODAF products, software views and architecture toolset
- Mission threads cross contractor/system/build boundaries, sparse on quality attributes
- SoS requirements often too vague
- Black box nature of many components
- Architecture/Design documentation does not match implementation

Programmatic

- Multiple Contractors' architectural products (views) for systems/software to be integrated into many weapons/sensor platforms and SILs
- Concurrent Development of system and software architectures – tension between system and software architects
- Life-cycle of many builds with spin-outs to the field from each build



